A User-Friendly Web Tool for School Timetabling

Shana Van Dessel Joost Vennekens

Department of Computer Science, KU Leuven
Technology Campus De Nayer, Sint-Katelijne-Waver Belgium

1 Timetabling in secondary schools

The school timetabling problem (STP) is still a relevant problem for most secondary schools despite the amount of research being done on the subject and the variety of scheduling tools that are commercially available on the market [2]. Making timetables by hand is hard (especially for big(ger) schools). It requires sufficient manpower and a lot of time, which most secondary schools do not have. Besides, the manually created schedules are usually far from ideal. Therefore most schools use scheduling tools to make their timetables. The most common ones are Untis and EDT. Although making use of these tools is a big improvement over the manual work, they do not meet all the requirements of the user. The most important problem is the lack of user-friendliness. The existing tools allow very optimal timetables (according to some criteria), provided the user is able to properly work with the tool. But in most cases the user is not even able to create a complete timetable (using the tool), mostly because of the difficult interface which does allow a wide range of restrictions, but does not give a proper overview and the needed explanation when the tool has troubles with one or more of the restrictions. Instead of using the tool at its full potential, the user just uses it to make a set of initial timetables and then adjusts these (manually) further to his wishes.

2 IDP-based web tool

Our goal is to create a more user-friendly tool for solving the STP, by means of a declarative specification of the STP instance. On the one hand we used an existing prototype for an STP-solving application, implemented in IDP (Imperative Declarative Programming) and on the other hand we created an easy-to-use web tool as interface between the user and the IDP application.

2.1 IDP application

So the application from which we start, was created in IDP (Imperative Declarative Programming). IDP is a Knowledge Base System (KBS) especially developed for the FO(.) language (an extension of first-order logic) [1], which means it uses a declarative specification of knowledge.

For example:

```
! g c : #{ l : Lesson(l,g) = c} = Hours(Grade(g),c).
```

is one of the IDP rules from the existing application. It says that for each (!) class group **g** and course **c**, the sum (#) of the number of lessons **l** each class group **g** follows a lesson **l** of course **c**, must be equal to the number of hours the grade (the class group **g** is in) has to follow the course **c**.

In addition, IDP is an implementation of the Knowledge Base Paradigm so it allows for different inference tasks to execute on the same knowledge base.

2.2 Web tool

Our web tool has four major tasks. It needs to communicate with the user, both input and output data, and with the IDP application, also both input and output. For developing the web application we needed an application server for all the logic and a database for storing the school's data. We have chosen a Glassfish server and a MySQL database (both open source software). We used a computing platform to ease our work, namely Java EE (Java platform Enterprise Edition), it is an ideal platform for developing applications (largely because of the big cohesion between all the different components (web pages, server, database)). First we need to collect all data (needed by the IDP application) from the user. For this we used CSV (comma-separated-values), a simple data format in which most school's already have their data. The user needs to upload three CSV files with the following data; All direction and years, all courses in the different directions and how many hours each course is taught and finally all classes in the different directions and the number of students in each class. We parse these CSV files and upload the data to our database. The user is also able to change every piece of data using an easy GUI (graphical user interface) consisting of text fields and buttons. Then we need to convert our information to the right format for the IDP application and write it to an IDP (input) file, so we were able to run the application with the user's data. We choose to let the application write its results to an IDP file. Next we read the IDP output file, convert this data by parsing it and uploading it to our database. Finally we write the resulting data to a CSV file and return this file to the user as a download. The output consists of a list for each class with the class hours (equal to the number of lessons in one week) in combination with the different courses taught to this class.

3 Conclusion and future work

We created a web tool that is easy to use with a limited amount of information (without any courses or manuals) and interacts with an IDP application in a way that is transparent for the user. Our web tool is still a prototype, that is not yet able to create practical timetables. But after evaluating our web tool in collaboration with a secondary school, we came to the conclusion that there is great potential for IDP-based applications if we provide a user-friendly interface (to shield the IDP-specific things from the user). The IDP application as well as the web tool need to be improved, but the big advantage is that both can easily be extended with additional functions so our web tool would be able to generate practical timetables for secondary schools in no time. The less advanced functions of our tool are well compensated by the user-friendliness of it and the extent to which the tool is ready to use without courses and manuals.

References

[1] Maurice Bruynooghe, Hendrik Blockeel, Bart Bogaerts, Broes De Cat, Stef De Pooter, Joachim Jansen, Anthony Labarre, Jan Ramon, Marc Denecker, and Sicco Verwer. Predicate logic as a modeling language: Modeling and solving some machine learning and data mining problems with IDP3. *Theory and Practice of Logic Programming*, 2014.

[2] Nelishia Pillay. A survey of school timetabling research. *Annals of Operations Research*, 218(1):261–293, July 2014.